

## CLAIMS

What is claimed is:

1. A processor, comprising:

a processing core that generates memory addresses to access a memory and on which a plurality of methods operate;

a cache coupled to said processing core, said cache used to store sets of local variables, each set being used by a method; and

a programmable register containing a pointer to a currently active method's set of local variable;

wherein said cache includes at least two sets of local variables corresponding to different methods including a calling method that calls a called method, wherein the set of local variables corresponding to the called method is mapped adjacent a pointer to the set of local variables corresponding to the calling method.

2. The processor of claim 1 further including two or more sets of local variables corresponding to two or more nested methods, each set of local variables corresponding to a calling method separated from the set of local variables corresponding to a called method by a pointer to the set of local variables corresponding to the calling method.

3. The processor of claim 1 further including three or more sets of local variables corresponding to three or more nested methods, each set of local variables corresponding to a calling method separated from the set of local variables corresponding to a called method by a pointer to the set of local variables corresponding to the calling method.

4. The processor of claim 1 wherein, upon completion of a method, the pointer to the local variables associated with the calling method is retrieved by the processing core subtracting an offset value from the pointer associated with the called method to compute a modified pointer value and retrieving the value from the cache at the location pointed by the modified pointer value.
5. The processor of claim 1 wherein the pointer to a newly called method is calculated by the processor core adding an offset value and a local variable size value to a pointer associated with the calling method.
6. A method of storing local variables, comprising:
  - storing a first local variable set in a cache, the first local variable set associated with a first method and having a first pointer that identifies an initial variable in the first local variable set;
  - storing the first pointer on top of the first local variable set when the first method calls a second method; and
  - mapping a second local variable set associated with the second method adjacent the first pointer;wherein the first method calls the second method.

7. The method of claim 6 further comprising storing in the cache, adjacent the first local variable set and first pointer, a second local variable set associated with the second method and having a second pointer that identifies an initial variable in the second local variable set.
8. The method of claim 7 wherein, upon finishing the second method, the first pointer is retrieved from the cache by subtracting an offset from the second pointer.
9. The method of claim 6 wherein storing the first pointer includes adding an offset and a local variable size to location pointed by the first pointer.
10. A cache subsystem, comprising:
  - a cache controller; and
  - cache memory coupled to said cache controller, said cache memory adapted to store a plurality of sets of local variables, each set being used by a separate method, wherein each of a plurality of sets of local variables is mapped adjacent a local variable pointer associated with a calling method.
11. The cache subsystem wherein said cache memory is adapted to store at least three sets of local variables, each set used by a different method and at least two of said sets mapped adjacent a local variable pointer associated with a calling method.
12. The cache subsystem of claim 10 wherein a pointer value is retrieved upon completion of a method by subtracting an offset from a pointer associated with the completed method.

13. A processor, comprising:
  - a processing core that generates memory addresses to access a memory and on which a plurality of methods operate;
  - a means for storing a plurality of local variable sets, each set associated with a method that runs on the core and each set having a base pointer, and for mapping a set of local variables associated with a called method adjacent a pointer value associated with the method that called the called method.
14. The processor of claim 13 wherein the means further comprise a means for subtracting an offset value from the pointer associated with the called method to compute a modified pointer value and for retrieving the value at a location pointed to by the modified pointer value.
15. The processor of claim 13 wherein the means further comprises a means for adding an offset value and a local variable size value to a pointer associated with the calling method.